

# Comparison of Factorization-based Filtering for Landing Navigation

James S. McCabe<sup>\*</sup>   Aaron J. Brown<sup>†</sup>   Kyle J. DeMars<sup>‡</sup>   John M. Carson III<sup>§</sup>

This paper develops and analyzes methods for fusing inertial navigation data with external data, such as data obtained from an altimeter and a star camera. The particular filtering techniques are based upon factorized forms of the Kalman filter, specifically the UDU and Cholesky factorizations. The factorized Kalman filters are utilized to ensure numerical stability of the navigation solution. Simulations are carried out to compare the performance of the different approaches along a lunar descent trajectory using inertial and external data sources. It is found that the factorized forms improve upon conventional filtering techniques in terms of ensuring numerical stability for the investigated landing navigation scenario.

## I. Introduction

MINIMUM variance estimation is widely adopted among the scientific and engineering communities, as illustrated by the widespread use of the Kalman filter (KF)<sup>1</sup> and its variants, such as the extended Kalman filter (EKF).<sup>2,3</sup> The standard KF and EKF algorithms operate on the mean vector,  $\mathbf{m}$ , and covariance matrix,  $\mathbf{P}$ , where  $\mathbf{P}$  describes the statistics of the state,  $\mathbf{x}$ , which typically consists of position, velocity, attitude, and other parameters.

By definition,  $\mathbf{P}$  is a symmetric, positive definite matrix; these are conditions that must hold at every filtering step. While  $\mathbf{P}$  cannot lose symmetry or positive definiteness in a mathematical sense, they can be lost numerically due to roundoff errors, finite precision, or other such computer limitations. Symmetry is easily restored using brute-force symmetrization. Enforcing positive definiteness, however, is a much more demanding challenge. Positive definiteness can be lost in a situation where there is a large *a priori* uncertainty followed by precise measurements that can drive the *a posteriori* uncertainty close to zero. This scenario occurs commonly in landing navigation, where uncertainties naturally grow unabated for large periods of time before measurement data, such as altimetry, become available. Another situation is when the condition number of  $\mathbf{P}$ , i.e. the ratio of the largest to the smallest singular values of  $\mathbf{P}$ , is very large. An example of this is a filter that estimates both position and a sensor bias, in which there is a very large position uncertainty but a very small bias uncertainty. When  $\mathbf{P}$  is ill-conditioned in this manner, the mathematical operations performed on  $\mathbf{P}$  during filtering can cause it to lose positive definiteness.

A common approach to guarding against loss of positive definiteness is to introduce an underweighting factor that effectively softens the update when the combination of large prior uncertainties and precise measurements is encountered.<sup>4</sup> An alternative approach is to employ factorized formulations of the covariance matrix during the prediction and correction stages of the filter cycle. Formulations such as the Cholesky square-root factorization or the UDU factorization replace the full (i.e. unfactorized) covariance matrix with covariance matrix factors. As described in this paper, these formulations provide, at a minimum, a simple check for positive definiteness, thus obviating the performance degradation, or worse, filter divergence, that can result from a non-positive definite covariance matrix.

In the Cholesky factorization technique,  $\mathbf{P}$  is decomposed using a lower triangular matrix square root factor  $\mathbf{S}$  such that<sup>5</sup>

$$\mathbf{P} = \mathbf{S}\mathbf{S}^T.$$

---

<sup>\*</sup>Graduate Student, Missouri University of Science and Technology, Rolla, MO 65409, AIAA Student Member.

<sup>†</sup>Aerospace Engineer, NASA Johnson Space Center, Mail Code EG6, Houston, TX 77058.

<sup>‡</sup>Assistant Professor, Missouri University of Science and Technology, Rolla, MO 65409, AIAA Senior Member.

<sup>§</sup>Senior GN&C Engineer, NASA Johnson Space Center (on IPA detail from JPL), AIAA Associate Fellow.

Such a formulation not only guarantees symmetry, but can also guarantee the positive definiteness of  $\mathbf{P}$ . These benefits come at the cost of performance, as Cholesky factorization requires taking computationally expensive square roots. When computational throughput is not an issue, Cholesky-based methods provide clear advantages in numerical precision and robustness over working directly with the covariance matrix. When throughput is an issue, however, such as onboard a spacecraft flight computer, Cholesky factorization may not be feasible.

In such a situation, the UDU factorization<sup>6,7</sup> is a desirable alternative. The UDU factorization provides high degrees of precision and robustness, while not paying the computational burdens of the Cholesky factorization. In the UDU factorization,  $\mathbf{P}$  is decomposed using two matrix factors; an upper-diagonal matrix with ones along the diagonal (henceforth called a unit upper-diagonal matrix),  $\mathbf{U}$ , and a diagonal matrix,  $\mathbf{D}$ , such that

$$\mathbf{P} = \mathbf{U}\mathbf{D}\mathbf{U}^T.$$

The UDU factorization provides several numerical advantages. First, like the Cholesky factorization, the nature of the UDU factorization guarantees that  $\mathbf{P}$  is symmetric. Second, positive definiteness of the covariance matrix is easily determined by simply examining the entries of  $\mathbf{D}$ . If the entries are all positive, then  $\mathbf{P}$  is positive definite. UDU by itself does not *guarantee* positive definiteness, but it provides a very easy way to check for it. Third, UDU does not require taking square roots, which, as shown by Bierman [6, p. 82-90], results in a more computationally efficient algorithm than Cholesky.

This paper develops and applies Cholesky- and UDU-based factorized formulations of the extended Kalman filter for the problem of incorporating high-rate inertial measurement unit (IMU) data with external sensor data for the problem of navigating during a descent-to-landing scenario. Section II provides an overview of the dynamics modeling, including IMU modeling. Section III presents the sensor models used for simulating external data. Section IV develops the factorized filters considered in this work, beginning with a description of the multiplicative extended Kalman filter (MEKF). Section V demonstrates the application of the MEKF and factorized formulations of the MEKF to a lunar descent navigation problem. Section VI provides conclusions for the work presented in this paper.

## II. Dynamics Modeling

The continuous equations of motion for a vehicle with the aid of a strapdown IMU are given by<sup>8</sup>

$$\begin{aligned}\dot{\mathbf{r}}_{\text{imu}}^i(t) &= \mathbf{v}_{\text{imu}}^i(t) \\ \dot{\mathbf{v}}_{\text{imu}}^i(t) &= \mathbf{a}_g^i(\mathbf{r}_{\text{imu}}^i(t) + \mathbf{T}_c^i(t)\mathbf{r}_{\text{cg/imu}}^c) + \mathbf{T}_c^i(t)\mathbf{a}_{\text{ng}}^c(t) \\ \dot{\bar{\mathbf{q}}}_i^c(t) &= \frac{1}{2}\bar{\boldsymbol{\omega}}_{c/i}^c(t) \otimes \bar{\mathbf{q}}_i^c(t),\end{aligned}$$

where superscript  $i$  denotes the inertial frame, superscript  $c$  denotes the IMU case frame,  $\mathbf{r}_{\text{imu}}$  is the position of the IMU,  $\mathbf{v}_{\text{imu}}$  is the velocity of the IMU,  $\mathbf{r}_{\text{cg/imu}}^c$  is the position of the center of gravity (CG) with respect to the IMU,  $\mathbf{a}_g(\cdot)$  is the gravitational acceleration,  $\mathbf{a}_{\text{ng}}$  is the non-gravitational acceleration,  $\bar{\mathbf{q}}_i^c$  is the quaternion representing the orientation of the IMU case frame with respect to the inertial frame,  $\mathbf{T}_c^i$  is the rotation matrix representing the orientation of the inertial frame with respect to the IMU case frame,  $\boldsymbol{\omega}_{c/i}$  is the angular velocity of the IMU case frame with respect to the inertial frame,  $\bar{\boldsymbol{\omega}}$  is the pure quaternion formed from the vector  $\boldsymbol{\omega}$ , and  $\otimes$  represents quaternion multiplication defined such that the order of the quaternions is the same as the order of the equivalent rotation matrices. To simplify the nomenclature, let

$$\begin{aligned}\mathbf{r}_{\text{imu}}^i(t) &\rightarrow \mathbf{r}(t), \quad \mathbf{v}_{\text{imu}}^i(t) \rightarrow \mathbf{v}(t), \quad \mathbf{a}_g^i(\cdot) \rightarrow \mathbf{g}(\cdot), \quad \mathbf{T}_c^i(t) \rightarrow \mathbf{T}^T(t), \quad \mathbf{a}_{\text{ng}}^c(t) \rightarrow \mathbf{a}(t), \\ \bar{\mathbf{q}}_i^c(t) &\rightarrow \bar{\mathbf{q}}(t), \quad \boldsymbol{\omega}_{c/i}^c(t) \rightarrow \boldsymbol{\omega}(t), \quad \mathbf{r}_{\text{cg/imu}}^c \rightarrow \mathbf{d}, \quad \text{and} \quad \mathbf{r}_{\text{imu}}^i(t) + \mathbf{T}_c^i(t)\mathbf{r}_{\text{cg/imu}}^c \rightarrow \mathbf{s}(t).\end{aligned}$$

With these substitutions, the continuous-time, nonlinear, dynamical system governing the evolution of the position, velocity, and attitude of a vehicle under the influence of gravitational and non-gravitational acceleration, as well as angular velocity motion, is succinctly written as

$$\dot{\mathbf{r}}(t) = \mathbf{v}(t) \tag{1a}$$

$$\dot{\mathbf{v}}(t) = \mathbf{g}(\mathbf{s}(t)) + \mathbf{T}^T(t)\mathbf{a}(t) \tag{1b}$$

$$\dot{\bar{\mathbf{q}}}(t) = \frac{1}{2}\bar{\boldsymbol{\omega}}(t) \otimes \bar{\mathbf{q}}(t). \tag{1c}$$

## A. IMU Modeling

IMUs can operate in several ways, such as returning non-gravitational acceleration and angular velocity measurements or returning integrated non-gravitational acceleration and integrated angular velocity measurements. Additionally, internal compensation techniques can be employed to account for coning, sculling, and scrolling effects.<sup>9</sup> In all cases, the output of the IMU is corrupted by a variety of error sources, which include, but are not limited to: startup bias, walking biases due to bias instability, thermo-mechanical noise, scale factor errors, axes misalignments, nonorthogonality of the axes, and quantization.<sup>10</sup> In the current work, only the effects of stationary biases and thermo-mechanical noise are considered; additionally, it is assumed that the integrated non-gravitational acceleration and angular velocity are output by the IMU. This yields the IMU model for this work as

$$\Delta \mathbf{v}_{m,k} = \Delta \mathbf{v}_k + \mathbf{b}_v + \mathbf{w}_{v,k} \quad (2a)$$

$$\Delta \boldsymbol{\theta}_{m,k} = \Delta \boldsymbol{\theta}_k + \mathbf{b}_\theta + \mathbf{w}_{\theta,k}, \quad (2b)$$

where  $\Delta \mathbf{v}_k$  is the true integrated non-gravitational acceleration in the IMU case frame,  $\mathbf{b}_v$  is the accelerometer bias,  $\mathbf{w}_{v,k}$  is the accelerometer noise,  $\Delta \boldsymbol{\theta}_k$  is the true integrated angular velocity of the IMU case frame with respect to the inertial frame and expressed in the IMU case frame,  $\mathbf{b}_\theta$  is the gyro bias, and  $\mathbf{w}_{\theta,k}$  is the gyro noise. It is assumed herein that the biases and noises are all zero mean and mutually independent, with covariances defined by

$$\mathbb{E}\{\mathbf{b}_v \mathbf{b}_v^T\} = \mathbf{B}_v, \quad \mathbb{E}\{\mathbf{b}_\theta \mathbf{b}_\theta^T\} = \mathbf{B}_\theta, \quad \mathbb{E}\{\mathbf{w}_{v,k} \mathbf{w}_{v,\ell}^T\} = \mathbf{W}_{v,k} \delta_{k,\ell}, \quad \text{and} \quad \mathbb{E}\{\mathbf{w}_{\theta,k} \mathbf{w}_{\theta,\ell}^T\} = \mathbf{W}_{\theta,k} \delta_{k,\ell},$$

where  $\delta_{k,\ell}$  represents the Kronecker delta; that is  $\delta_{k,\ell} = 1$  for  $k = \ell$ , and zero otherwise. Said another way, the accelerometer and gyro noises are taken to be white noises.

It is oftentimes advantageous to “invert” Eqs. (2) in order to solve for the true integrated non-gravitational acceleration and angular velocity. When the IMU is modeled with a more complete error structure, this process is non-trivial; however, for the case at hand, the inversion process yields

$$\Delta \mathbf{v}_k = \Delta \mathbf{v}_{m,k} - \mathbf{b}_v - \mathbf{w}_{v,k}$$

$$\Delta \boldsymbol{\theta}_k = \Delta \boldsymbol{\theta}_{m,k} - \mathbf{b}_\theta - \mathbf{w}_{\theta,k},$$

from which one may obtain an estimate of the true integrated values as  $\Delta \hat{\mathbf{v}}_k = \mathbb{E}\{\Delta \mathbf{v}_k\}$  and  $\Delta \hat{\boldsymbol{\theta}}_k = \mathbb{E}\{\Delta \boldsymbol{\theta}_k\}$ , which gives

$$\Delta \hat{\mathbf{v}}_k = \Delta \mathbf{v}_{m,k} - \hat{\mathbf{b}}_{v,k} \quad (3a)$$

$$\Delta \hat{\boldsymbol{\theta}}_k = \Delta \boldsymbol{\theta}_{m,k} - \hat{\mathbf{b}}_{\theta,k}, \quad (3b)$$

where the zero-mean nature of the noise is taken into account.

## B. Discretized Dynamics

As IMU data is typically acquired at a high sampling rate, it is typical to convert the continuous-time dynamical system model given by Eqs. (1) into a discrete-time dynamical system model. Thus, the assumption that the non-gravitational acceleration and angular velocity are approximately constant over a small time step is introduced; that is,

$$\mathbf{a}_k = \frac{\Delta \mathbf{v}_k}{\Delta t_k} \quad \text{and} \quad \boldsymbol{\omega}_k = \frac{\Delta \boldsymbol{\theta}_k}{\Delta t_k}.$$

Applying analytical integration techniques to Eqs. (1) under the assumption of constant non-gravitational acceleration and angular velocity, it can be shown that the position, velocity, and attitude evolve according to

$$\mathbf{r}_k = \mathbf{r}_{k-1} + \mathbf{v}_{k-1} \Delta t_k + \frac{1}{2} \mathbf{T}_{k-1}^T \left( \mathbf{I}_{3 \times 3} + \frac{1}{3} [\Delta \boldsymbol{\theta}_k \times] \right) \Delta \mathbf{v}_k \Delta t_k + \frac{1}{2} \left( \mathbf{g}_{k-1} - \frac{1}{3} \mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [d \times] \Delta \boldsymbol{\theta}_k \right) \Delta t_k^2 \quad (4a)$$

$$\mathbf{v}_k = \mathbf{v}_{k-1} + \mathbf{T}_{k-1}^T \left( \mathbf{I}_{3 \times 3} + \frac{1}{2} [\Delta \boldsymbol{\theta}_k \times] \right) \Delta \mathbf{v}_k + \left( \mathbf{g}_{k-1} - \frac{1}{2} \mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [d \times] \Delta \boldsymbol{\theta}_k \right) \Delta t_k \quad (4b)$$

$$\bar{\mathbf{q}}_k = \bar{\mathbf{q}}(\Delta \boldsymbol{\theta}_k) \otimes \bar{\mathbf{q}}_{k-1}, \quad (4c)$$

where  $\mathbf{G}_{k-1}$  is the Jacobian of the gravitational acceleration function and  $[\cdot \times]$  represents the skew-symmetric cross product matrix.

Equations (4) represent the nonlinear dynamical system for the forward evolution of the position, velocity, and attitude of a vehicle as defined by the integrated non-gravitational acceleration and angular velocity experienced by the vehicle, where the accelerometer model parameters (bias, noise, etc.) and the gyro model parameters (bias, noise, etc.) are taken into account. Additionally, Eqs. (4) makes use of a model of the gravitational acceleration, where an offset between the CG and IMU locations can be naturally included.

In addition to the position, velocity, and attitude of the vehicle, there are typically other dynamical quantities of interest, such as IMU and other sensor biases. In the present work, these other parameters are taken to have identity dynamics, such that, if  $\mathbf{p}_k$  represents these parameters at time  $t_k$ , the discrete-time dynamics are

$$\mathbf{p}_k = \mathbf{p}_{k-1}. \quad (5)$$

It is, of course, possible to interchange this dynamical system with another one. One common practice would be to assume that the parameters follow an exponentially correlated random variable (ECRV) model.<sup>3</sup>

### C. Linearized Error Dynamics

The filtering methods employed in this work are all predicated upon the application of linearization to handle nonlinear dynamics and measurements, which leads to the class of Kalman filters commonly called extended Kalman filters. In this approach, the state estimates are propagated by integration of the equations of motion with the dynamics evaluated at the current state estimate. Therefore, the estimates of position, velocity, and attitude are taken to have dynamics that are given by

$$\dot{\hat{\mathbf{r}}}(t) = \hat{\mathbf{v}}(t) \quad (6a)$$

$$\dot{\hat{\mathbf{v}}}(t) = \mathbf{g}(\hat{\mathbf{s}}(t)) + \hat{\mathbf{T}}^T(t)\hat{\mathbf{a}}(t) \quad (6b)$$

$$\dot{\hat{\mathbf{q}}}(t) = \frac{1}{2}\hat{\boldsymbol{\omega}}(t) \otimes \hat{\mathbf{q}}(t). \quad (6c)$$

As with the true dynamics of the system, it is desired to find a discretized form of the dynamics for the estimated states based on the fact that high-rate IMU data is employed in propagating the state estimate. Therefore, similar to the true dynamics case, it is assumed that the estimated non-gravitational acceleration and estimated angular velocity are constant over a small time step, or

$$\hat{\mathbf{a}}_k = \frac{\Delta \hat{\mathbf{v}}_k}{\Delta t_k} \quad \text{and} \quad \hat{\boldsymbol{\omega}}_k = \frac{\Delta \hat{\boldsymbol{\theta}}_k}{\Delta t_k}.$$

With this assumption, applying analytical integration to Eqs. (6) yields the discretized dynamics for the estimated position, velocity, and attitude as

$$\hat{\mathbf{r}}_k = \hat{\mathbf{r}}_{k-1} + \hat{\mathbf{v}}_{k-1}\Delta t_k + \frac{1}{2}\hat{\mathbf{T}}_{k-1}^T \left( \mathbf{I}_{3 \times 3} + \frac{1}{3}[\Delta \hat{\boldsymbol{\theta}}_k \times] \right) \Delta \hat{\mathbf{v}}_k \Delta t_k + \frac{1}{2} \left( \hat{\mathbf{g}}_{k-1} - \frac{1}{3}\hat{\mathbf{G}}_{k-1}\hat{\mathbf{T}}_{k-1}^T[\hat{\mathbf{d}} \times] \Delta \hat{\boldsymbol{\theta}}_k \right) \Delta t_k^2 \quad (7a)$$

$$\hat{\mathbf{v}}_k = \hat{\mathbf{v}}_{k-1} + \hat{\mathbf{T}}_{k-1}^T \left( \mathbf{I}_{3 \times 3} + \frac{1}{2}[\Delta \hat{\boldsymbol{\theta}}_k \times] \right) \Delta \hat{\mathbf{v}}_k + \left( \hat{\mathbf{g}}_{k-1} - \frac{1}{2}\hat{\mathbf{G}}_{k-1}\hat{\mathbf{T}}_{k-1}^T[\hat{\mathbf{d}} \times] \Delta \hat{\boldsymbol{\theta}}_k \right) \Delta t_k \quad (7b)$$

$$\hat{\mathbf{q}}_k = \hat{\mathbf{q}}(\Delta \hat{\boldsymbol{\theta}}_k) \otimes \hat{\mathbf{q}}_{k-1}, \quad (7c)$$

where  $\Delta \hat{\mathbf{v}}_k$  and  $\Delta \hat{\boldsymbol{\theta}}_k$  are computed from Eqs. (3). The corresponding dynamics governing the evolution of the estimated parameters,  $\hat{\mathbf{p}}_k$ , are

$$\hat{\mathbf{p}}_k = \hat{\mathbf{p}}_{k-1}. \quad (8)$$

In order to determine a set of linearized error dynamics, it is first necessary to define the error. The error is generally defined as the difference between the true quantity and its estimate. This definition works naturally for defining position, velocity, and parameter errors, such that

$$\mathbf{e}_{r,k} = \mathbf{r}_k - \hat{\mathbf{r}}_k \quad (9a)$$

$$\mathbf{e}_{v,k} = \mathbf{v}_k - \hat{\mathbf{v}}_k \quad (9b)$$

$$\mathbf{e}_{p,k} = \mathbf{p}_k - \hat{\mathbf{p}}_k. \quad (9c)$$

Simple subtraction, however, is not directly applicable to the attitude, which ultimately stems from the fact that attitude is represented by a quaternion. Instead, the attitude error is defined as

$$\mathbf{e}_{\phi,k} = 2 \cdot \text{vec}(\bar{\mathbf{q}}_k \otimes \hat{\mathbf{q}}_k^{-1}), \quad (10)$$

where  $\hat{\mathbf{q}}_k^{-1}$  is the inverse of the quaternion  $\hat{\mathbf{q}}_k$ , and the  $\text{vec}(\cdot)$  operator extracts the vector part of the input quaternion. This definition is effectively a small-angle assumption on the attitude error.

With the errors defined as in Eqs. (9) and (10), the error dynamics are determined by substituting Eqs. (4), (5), (7), and (8) into Eqs. (9) and (10). The resulting expressions are then reduced by the application of first-order Taylor series expansions to determine the set of linearized error dynamics, which can be represented in the form

$$\mathbf{e}_k = \mathbf{F}_{k-1} \mathbf{e}_{k-1} + \mathbf{M}_{k-1} \mathbf{w}_{k-1}, \quad (11)$$

where  $\mathbf{e}_k^T = [\mathbf{e}_{r,k}^T \ \mathbf{e}_{v,k}^T \ \mathbf{e}_{\phi,k}^T \ \mathbf{e}_{p,k}^T]$ ,  $\mathbf{F}_{k-1}$  is the dynamics Jacobian,  $\mathbf{w}_{k-1}^T = [\mathbf{w}_{v,k-1}^T \ \mathbf{w}_{\theta,k-1}^T]$ , and  $\mathbf{M}_{k-1}$  is a shaping matrix. It should be noted that, if  $n+1$  is the total dimension of  $[\mathbf{r}_k^T \ \mathbf{v}_k^T \ \bar{\mathbf{q}}_k^T \ \mathbf{p}_k^T]$ , then the dimension of  $\mathbf{F}_{k-1}$  is  $n \times n$ . Additionally, if  $m$  is the dimension of  $\mathbf{w}_{k-1}$ , then the dimension of  $\mathbf{M}_{k-1}$  is  $n \times m$ .

### III. Observation Modeling

Two sensors are modeled as generating imperfect, partial measurements regarding the state of the system: a spherical altimeter and a quaternion star camera. These sensors are chosen to represent the case where a “minimal” set of external information is provided. Additionally, the models presented here are low fidelity in that effects such as the terrain or the region of the sky imaged by the star tracker are not considered.

#### A. Altimeter Modeling

The altimeter is modeled as the range to a spherical surface along the nadir direction from the altimeter’s location onboard the vehicle with corruption of the true signal occurring due to random bias and noise; this yields the model

$$z_k = (\|\mathbf{r}_{\text{alt},k}^i\| - r_{\text{sph}}) + b_{\text{alt}} + v_{\text{alt},k},$$

where  $\mathbf{r}_{\text{alt},k}$  is the position of the altimeter with respect to the planet center,  $r_{\text{sph}}$  is the spherical planet radius,  $b_{\text{alt}}$  is a zero-mean, constant, random bias, and  $v_{\text{alt},k}$  is a zero-mean, white-noise sequence. The covariances of the bias and noise are given, respectively, by

$$\text{E}\{b_{\text{alt}}^2\} = B_{\text{alt}} \quad \text{and} \quad \text{E}\{v_{\text{alt},k} v_{\text{alt},\ell}\} = V_{\text{alt}} \delta_{k,\ell}.$$

The position of the altimeter is related to the position of the IMU through the relationship

$$\mathbf{r}_{\text{alt},k}^i = \mathbf{r}_{\text{imu},k}^i + \mathbf{T}_{c,k}^i \mathbf{r}_{\text{alt}/\text{imu}}^c.$$

Akin to the developments of Section II-C, filtering methods based on linearization evaluate an estimated measurement by evaluating the measurement function at the current state estimate. For the altimeter measurement, this yields

$$\hat{z}_k = (\|\hat{\mathbf{r}}_{\text{alt},k}^i\| - r_{\text{sph}}) + \hat{b}_{\text{alt},k},$$

where  $\hat{\mathbf{r}}_{\text{alt},k}^i$  is the estimated position of the altimeter, which is

$$\hat{\mathbf{r}}_{\text{alt},k}^i = \hat{\mathbf{r}}_{\text{imu},k}^i + \hat{\mathbf{T}}_{c,k}^i \mathbf{r}_{\text{alt}/\text{imu}}^c.$$

Defining the deviation of the true measurement from the estimated measurement to be

$$\delta z_k = z_k - \hat{z}_k$$

and substituting the previously established relationships for  $z_k$  and  $\hat{z}_k$ , it can be shown that, to first order,

$$\delta z_k = \mathbf{H}_{r,k} \mathbf{e}_{r,k} + \mathbf{H}_{\phi,k} \mathbf{e}_{\phi,k} + \mathbf{H}_{b,k} e_{b_{\text{alt}},k} + v_{\text{alt},k}, \quad (12)$$

where  $\mathbf{e}_{r,k} = \mathbf{r}_{\text{imu},k}^i - \hat{\mathbf{r}}_{\text{imu},k}^i$ ,  $\mathbf{e}_{\phi,k} = 2 \cdot \text{vec}(\bar{\mathbf{q}}_{i,k}^c \otimes (\hat{\mathbf{q}}_{i,k}^c)^{-1})$ , and  $e_{b_{\text{alt}},k} = b_{\text{alt}} - \hat{b}_{\text{alt},k}$ . The elements in this first-order expansion yield the Jacobian of the altimeter measurement model, which will be employed in the filtering schemes discussed in Section IV.

## B. Star Camera Modeling

The star camera is modeled to be the orientation of a frame attached to the star camera, denoted the star camera (sc) frame, with respect to the inertial frame with corruption of the true signal occurring due to random bias and noise; this model is represented by

$$\bar{\mathbf{z}}_k = \bar{\mathbf{q}}_{\text{err},k} \otimes \bar{\mathbf{q}}_c^{\text{sc}} \otimes \bar{\mathbf{q}}_{i,k}^c,$$

where  $\bar{\mathbf{q}}_{i,k}^c$  is the true attitude quaternion describing the orientation of the IMU case frame with respect to the inertial frame,  $\bar{\mathbf{q}}_c^{\text{sc}}$  is the attitude quaternion describing the orientation of the star camera frame with respect to the IMU case frame, and  $\bar{\mathbf{q}}_{\text{err}}$  is the quaternion containing rotational corruption due to bias and noise, which is taken to be of the form

$$\bar{\mathbf{q}}_{\text{err},k} = \begin{bmatrix} \sin\left(\frac{1}{2}\|\boldsymbol{\theta}_{\text{err},k}\|\right) \frac{\boldsymbol{\theta}_{\text{err},k}}{\|\boldsymbol{\theta}_{\text{err},k}\|} \\ \cos\left(\frac{1}{2}\|\boldsymbol{\theta}_{\text{err},k}\|\right) \end{bmatrix} \quad \text{where} \quad \boldsymbol{\theta}_{\text{err},k} = \mathbf{b}_{sc} + \mathbf{v}_{sc,k}.$$

Here,  $\mathbf{b}_{sc}$  is a zero-mean, constant, random bias, and  $\mathbf{v}_{sc,k}$  is a zero-mean, white-noise sequence. The covariances of the bias and noise are given, respectively, by

$$\mathbb{E}\{\mathbf{b}_{sc}\mathbf{b}_{sc}^T\} = \mathbf{B}_{sc} \quad \text{and} \quad \mathbb{E}\{v_{sc,k}v_{sc,\ell}\} = \mathbf{V}_{sc}\delta_{k,\ell}.$$

As with the altimeter model, it is necessary to evaluate an estimated measurement, which is, in this case, constructed as

$$\hat{\mathbf{z}}_k = \hat{\mathbf{q}}_{\text{err},k} \otimes \bar{\mathbf{q}}_c^{\text{sc}} \otimes \hat{\mathbf{q}}_{i,k}^c,$$

where  $\hat{\mathbf{q}}_{i,k}^c$  is the estimated attitude quaternion describing the orientation of the IMU case frame with respect to the inertial frame and  $\hat{\mathbf{q}}_{\text{err},k}$  is given by

$$\hat{\mathbf{q}}_{\text{err},k} = \begin{bmatrix} \sin\left(\frac{1}{2}\|\hat{\boldsymbol{\theta}}_{\text{err},k}\|\right) \frac{\hat{\boldsymbol{\theta}}_{\text{err},k}}{\|\hat{\boldsymbol{\theta}}_{\text{err},k}\|} \\ \cos\left(\frac{1}{2}\|\hat{\boldsymbol{\theta}}_{\text{err},k}\|\right) \end{bmatrix} \quad \text{where} \quad \hat{\boldsymbol{\theta}}_{\text{err},k} = \hat{\mathbf{b}}_{sc,k}.$$

Unlike the altimeter, an additive deviation of the form

$$\delta\bar{\mathbf{z}}_k = \bar{\mathbf{z}}_k - \hat{\mathbf{z}}_k$$

is not proper since the attitude quaternion is not additive. Therefore, a multiplicative deviation, similar to how the attitude error is defined in Eq. (10), is defined as

$$\delta\boldsymbol{\psi}_k = 2 \cdot \text{vec}(\bar{\mathbf{z}}_k \otimes \hat{\mathbf{z}}_k^{-1}). \quad (13)$$

Then, by substituting for  $\bar{\mathbf{z}}_k$  and  $\hat{\mathbf{z}}_k$ , it can be shown that, to first order,

$$\delta\boldsymbol{\psi}_k = \mathbf{H}_{\phi,k}\mathbf{e}_{\phi,k} + \mathbf{H}_{b,k}\mathbf{e}_{b_{sc},k} + \mathbf{v}_{sc,k}, \quad (14)$$

where  $\mathbf{e}_{\phi,k} = 2 \cdot \text{vec}(\bar{\mathbf{q}}_{i,k}^c \otimes (\hat{\mathbf{q}}_{i,k}^c)^{-1})$  and  $\mathbf{e}_{b_{sc},k} = \mathbf{b}_{sc} - \hat{\mathbf{b}}_{sc,k}$ . The elements in this first-order expansion yield the Jacobian of the star camera measurement model, which will be employed in the filtering schemes discussed in Section IV.

## IV. Factorization-Based Filtering

Consider a system governed by the nonlinear discrete-time dynamics

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{M}_{k-1}\mathbf{w}_{k-1},$$

where  $\mathbf{x}_k$  is the state of the system (i.e. position, velocity, attitude, etc.) at time  $t_k$ ,  $\mathbf{f}(\cdot)$  represents the nonlinear dynamics,  $\mathbf{w}_{k-1}$  is a zero-mean, white process noise with covariance  $\mathbf{W}_{k-1}$ , and  $\mathbf{M}_{k-1}$  is a shaping matrix. Accompanying the nonlinear dynamical system are nonlinear measurements of the form

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k,$$

where  $\mathbf{z}_k$  is the collection of measurement data (i.e. altimetry, star camera, etc.) at time  $t_k$ ,  $\mathbf{h}(\cdot)$  is the nonlinear measurement function, and  $\mathbf{v}_k$  is a zero-mean, white measurement noise with covariance  $\mathbf{V}_k$ .

In the following developments, it is assumed that the state is comprised of the position of the IMU in the inertial frame ( $\mathbf{r}_k = \mathbf{r}_{\text{imu},k}^i$ ), the velocity of the IMU in the inertial frame ( $\mathbf{v}_k = \mathbf{v}_{\text{imu},k}^i$ ), the attitude of the IMU case frame with respect to the inertial frame ( $\bar{\mathbf{q}}_k = \bar{\mathbf{q}}_{i,k}^c$ ), and a collection of sensor biases ( $\mathbf{p}_k$ ), such that

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{r}_k \\ \mathbf{v}_k \\ \bar{\mathbf{q}}_k \\ \mathbf{p}_k \end{bmatrix} \quad \text{where} \quad \mathbf{p}_k = \begin{bmatrix} b_{v,k} \\ b_{\theta,k} \\ b_{\text{alt},k} \\ b_{sc,k} \end{bmatrix}.$$

### A. Multiplicative Extended Kalman Filter

The extended Kalman filter is ubiquitously used in nonlinear estimation, especially for the function of navigating vehicles in nonlinear environments. The EKF is typically broken into two stages: the predictor and the corrector. The predictor of the EKF propagates the mean and covariance of the state distribution between measurements, and the corrector updates the mean and covariance when new measurement data are received, such that the posterior covariance is minimized when a linear update is used. This is the minimum mean square error paradigm.

In the case of estimating a quaternion representation of the vehicle's attitude, however, additional machinery is required due to the facts that (a) in general, the sum of attitude quaternions is no longer a valid attitude quaternion and (b) constructing a covariance matrix to quantify the uncertainty in a quaternion estimate produces a singular covariance matrix.<sup>11</sup> Accordingly, the multiplicative extended Kalman filter is utilized.<sup>12</sup>

The cornerstone of the MEKF is that of a small angle assumption. This assumption permits the four-parameter quaternion description of attitude error to be described with three parameters, an attitude error vector, according to Eq. (10). Under this assumption, in contrast to the EKF, the covariance is used to statistically describe the three-parameter attitude error while maintaining the traditional error representation for the non-quaternion states; ultimately, this results in a non-singular representation for the covariance matrix.

Given the posterior mean and covariance at time  $t_{k-1}$  as  $\mathbf{m}_{k-1}^+$  and  $\mathbf{P}_{k-1}^+$ , respectively, the mean and covariance are propagated according to

$$\mathbf{m}_k^- = \mathbf{f}(\mathbf{m}_{k-1}^+) \quad (15a)$$

$$\mathbf{P}_k^- = \mathbf{F}_{k-1} \mathbf{P}_{k-1}^+ \mathbf{F}_{k-1}^T + \mathbf{M}_{k-1} \mathbf{W}_{k-1} \mathbf{M}_{k-1}^T, \quad (15b)$$

where  $\mathbf{F}_{k-1}$  is the Jacobian of  $\mathbf{f}(\cdot)$ . The representations for  $\mathbf{F}_{k-1}$  and  $\mathbf{M}_{k-1}$  come from Eq. (11), and the nonlinear dynamics,  $\mathbf{f}(\cdot)$  come from Eqs. (7) and (8).

To update the mean and covariance in the MEKF, the residual is first computed as the difference between the received measurement and the predicted measurement, i.e

$$\mathbf{r}_k = \mathbf{z}_k - \mathbf{h}(\mathbf{m}_k^-). \quad (16)$$

It should be noted that, in the case of non-additive measurements, such as with the star camera described previously, an appropriate re-casting of the residual must be considered. For the star camera, this is given by Eq. (13). The Kalman gain is then found from

$$\mathbf{K}_k = \mathbf{C}_k \mathbf{Z}_k^{-1},$$

where  $\mathbf{C}_k$  is the cross-covariance between the state and the measurement and  $\mathbf{Z}_k$  is the residual (or innovations) covariance. For the EKF, the cross-covariance and the residual covariance are given by

$$\begin{aligned} \mathbf{C}_k &= \mathbf{P}_k^- \mathbf{H}_k^T \\ \mathbf{Z}_k &= \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{V}_k, \end{aligned}$$

where  $\mathbf{H}_k$  is the Jacobian of  $\mathbf{h}(\cdot)$ . For the altimeter and star camera measurements,  $\mathbf{H}_k$  comes from Eqs. (12) and (14), respectively. Finally, the mean and covariance are updated as

$$\mathbf{m}_k^+ = \mathbf{m}_k^- + \Delta \mathbf{x}_k \quad (17)$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{Z}_k \mathbf{K}_k^T, \quad (18)$$

where

$$\Delta \mathbf{x}_k = \mathbf{K}_k \mathbf{r}_k. \quad (19)$$

Again, as there is a quaternion present in the state vector, care must be taken when applying Eq. (17). The portion of the state that does not correspond to the attitude quaternion is updated according to Eq. (17). The quaternion update, however, is carried out using quaternion multiplication as

$$\hat{\mathbf{q}}_k^+ = \begin{bmatrix} \frac{1}{2} \Delta \phi_k^+ \\ 1 \end{bmatrix} \otimes \hat{\mathbf{q}}_k^-,$$

where  $\Delta \phi_k^+$  is the portion of  $\Delta \mathbf{x}_k$  corresponding to the location of the three-parameter attitude error. Finally, the resulting posterior quaternion is re-normalized.

While the MEKF is an extremely powerful method for estimating the state of the dynamical system, as noted in Section I, it can suffer from numerical issues surrounding the nature of the covariance matrix, particularly in enforcing its symmetry and positive definiteness. Cholesky square-root factorization or UDU factorization of the covariance matrix are two approaches to alleviating this problem, which gives rise to the square-root multiplicative extended Kalman filter (SR-MEKF) and the UDU multiplicative extended Kalman filter (UDU-MEKF), respectively. Recall from Section I that the SR-MEKF naturally enforces symmetry and is capable of guaranteeing positive definiteness, but requires taking square roots. The UDU-MEKF, on the other hand, does not require taking square roots, naturally enforces symmetry, and provides a simple check for positive definiteness, but does not enforce positive definiteness. In both of these implementations, the mechanics of handling the quaternion states is identical. The key difference between the SR-MEKF and the UDU-MEKF, as compared to the standard MEKF, is in how the covariance is represented and handled in both the predictor and corrector stages of the filter.

## B. Square-Root Multiplicative Extended Kalman Filter

Consider the task of predicting the mean and covariance at  $t_k$ , written as  $\mathbf{m}_k^-$  and  $\mathbf{P}_k^-$ , given the *a posteriori* mean and covariance at  $t_{k-1}$ , written as  $\mathbf{m}_{k-1}^+$  and  $\mathbf{P}_{k-1}^+$ . In covariance form, the relationships are described by Eqs. (15). The principle idea behind the SR-MEKF is to work with the Cholesky factorization of the covariance matrix. That is, given  $\mathbf{S}_{k-1}^+$ , which satisfies  $\mathbf{P}_{k-1}^+ = \mathbf{S}_{k-1}^+ (\mathbf{S}_{k-1}^+)^T$ , it is desired to determine  $\mathbf{S}_k^-$ , which satisfies  $\mathbf{P}_k^- = \mathbf{S}_k^- (\mathbf{S}_k^-)^T$ . The mean propagation equation remains the same, and the square-root covariance propagation employs a QR decomposition, such that<sup>13,14</sup>

$$\begin{aligned} \mathbf{m}_k^- &= \mathbf{f}(\mathbf{m}_{k-1}^+) \\ \mathbf{S}_k^- &= \text{qr} \{ [\mathbf{F}_{k-1} \mathbf{S}_{k-1}^+ \mid \mathbf{M}_{k-1} \mathbf{T}_{k-1}]^T \}^T, \end{aligned}$$

where  $\mathbf{T}_{k-1}$  is a (lower triangular) Cholesky factorization of the process noise covariance matrix, and  $\text{qr}\{\cdot\}$  represents a QR decomposition of the non-square input matrix.<sup>15</sup> The QR decomposition used throughout is taken to be the so-called “economy” form that returns an upper triangular square-root factor; that is, the decomposition returns only the upper triangular matrix  $\mathbf{R}$  that is then transposed to give the appropriate lower triangular result.

For the update stage of the SR-MEKF, the same techniques are applied to update the mean and square-root factor as are used for the propagation stage.<sup>13,14</sup> The mean update remains unchanged from the EKF update of Eq. (17). The only difference is in how the Kalman gain is computed, which will be discussed shortly. The covariance update, however, is replaced by a square-root factor update. First, the square-root factor of the residual covariance,  $\mathbf{Y}_k$ , and the cross-covariance are computed as

$$\begin{aligned} \mathbf{Y}_k &= \text{qr} \{ [\mathbf{H}_k \mathbf{S}_k^- \mid \mathbf{L}_k]^T \}^T \\ \mathbf{C}_k &= \mathbf{S}_k^- [\mathbf{H}_k \mathbf{S}_k^-]^T, \end{aligned}$$



where  $\mathbf{V}_k = \mathbf{L}_k \mathbf{L}_k^T$ . Then, a set of update factors is determined via

$$\mathbf{U}_k = \mathbf{C}_k(\mathbf{Y}_k^-)^T = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_p],$$

which gives the Kalman gain to be

$$\mathbf{K}_k = \mathbf{U}_k \mathbf{Y}_k^{-1}.$$

The final step is to compute the square-root factor update. This is done via a sequence of rank-1 downdates using the columns of  $\mathbf{U}_k$ . First, set  $\mathbf{A} = (\mathbf{S}_k^-)^T$ ; then, for  $i = 1, \dots, p$ ,  $\mathbf{A}$  is updated according to

$$\mathbf{A} = \text{cholupdate}\{\mathbf{A}, \mathbf{u}_i, -1\}.$$

The updated square-root factor is  $\mathbf{S}_k^+ = \mathbf{A}^T$ . As with the  $\text{qr}\{\cdot\}$  operation, it is assumed that  $\text{cholupdate}\{\cdot\}$  operates on upper-triangular matrices, so the lower-triangular *a priori* Cholesky factor is transposed to begin the operations, and the output of the rank-1 downdate sequence is transposed to recover a lower-triangular Cholesky factor.

### C. The UDU Multiplicative Extended Kalman Filter

As with the square-root formulation, the mean propagation equation remains the same, while the covariance propagation equation is transformed as follows. This development summarizes D'Souza [16, Sec. 3.1 & Appendix A], and the reader is directed there for further details. The development begins with the full covariance propagation equation in Eq. (15b). The UDU-MEKF works with a UDU factorization of the covariance matrix; that is, given a unit upper-diagonal matrix  $\mathbf{U}_{k-1}^+$  and a diagonal matrix  $\mathbf{D}_{k-1}^+$ , which satisfy  $\mathbf{P}_{k-1}^+ = \mathbf{U}_{k-1}^+ \mathbf{D}_{k-1}^+ (\mathbf{U}_{k-1}^+)^T$ , it is desired to determine  $\mathbf{U}_k^-$  and  $\mathbf{D}_k^-$  that satisfy  $\mathbf{P}_k^- = \mathbf{U}_k^- \mathbf{D}_k^- (\mathbf{U}_k^-)^T$ . Noting the aforementioned UDU factorization, Eq. (15b) can be written as

$$\begin{aligned} \mathbf{U}_k^- \mathbf{D}_k^- (\mathbf{U}_k^-)^T &= \mathbf{F}_{k-1}^+ \mathbf{U}_{k-1}^+ \mathbf{D}_{k-1}^+ (\mathbf{U}_{k-1}^+)^T (\mathbf{F}_{k-1}^+)^T + \mathbf{M}_{k-1} \mathbf{W}_{k-1} \mathbf{M}_{k-1}^T \\ &= \underbrace{\begin{bmatrix} \mathbf{F}_{k-1}^+ \mathbf{U}_{k-1}^+ & \mathbf{M}_{k-1} \end{bmatrix}}_{\triangleq \mathbf{Y}_k^-} \underbrace{\begin{bmatrix} \mathbf{D}_{k-1}^+ & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{k-1} \end{bmatrix}}_{\triangleq \tilde{\mathbf{D}}_k^-} \begin{bmatrix} \mathbf{F}_{k-1}^+ \mathbf{U}_{k-1}^+ & \mathbf{M}_{k-1} \end{bmatrix}^T \\ &= \mathbf{Y}_k^- \tilde{\mathbf{D}}_k^- (\mathbf{Y}_k^-)^T \\ &= \mathbf{Y}_k^- (\mathbf{T}_k^-)^{-T} \left[ (\mathbf{T}_k^-)^T \tilde{\mathbf{D}}_k^- (\mathbf{T}_k^-) \right] (\mathbf{T}_k^-)^{-1} (\mathbf{Y}_k^-)^T, \end{aligned} \quad (20)$$

where  $\mathbf{Y}_k^-$  is  $n \times (n+m)$ ,  $\tilde{\mathbf{D}}_k^-$  and  $\mathbf{T}_k^-$  are  $(n+m) \times (n+m)$ , and  $\mathbf{T}_k^-$  is to be determined. Comparing the left-hand and right-hand sides of Eq. (20), it is desired to find  $\mathbf{T}_k^-$  such that

1.  $\mathbf{Y}_k^- (\mathbf{T}_k^-)^{-T} = [\mathbf{U}_k^- \quad \mathbf{0}_{n \times m}]$ , where  $\mathbf{U}_k^-$  is unit upper-diagonal, and
2.  $(\mathbf{T}_k^-)^T \tilde{\mathbf{D}}_k^- (\mathbf{T}_k^-)$  is diagonal.

Modified Weighted Gram-Schmidt (MWGS) orthogonalization is employed to satisfy these conditions by creating a set of orthogonal basis vectors  $\mathbf{b}_i$ ,  $i = 1, \dots, n+m$  that are functions of  $\mathbf{Y}_k^-$  and  $\tilde{\mathbf{D}}_k^-$ .<sup>a</sup> These vectors are then used to form  $\mathbf{T}_k^-$  as

$$\mathbf{T}_k^- = [\mathbf{b}_1 \quad \mathbf{b}_2 \quad \mathbf{b}_3 \quad \cdots \quad \mathbf{b}_n \quad \mathbf{b}_{n+1} \quad \cdots \quad \mathbf{b}_{n+m}]_{(n+m) \times (n+m)}.$$

If  $\mathbf{T}_k^-$  is partitioned as  $\mathbf{T}_k^- = [\mathbf{T}_{1k}^- \quad \mathbf{T}_{2k}^-]$ , where  $\mathbf{T}_{1k}^-$  is formed from the first  $n$  columns of  $\mathbf{T}_k^-$  and  $\mathbf{T}_{2k}^-$  is formed from the last  $m$  columns of  $\mathbf{T}_k^-$ , then Eq. (20) becomes

$$\begin{aligned} \mathbf{U}_k^- \mathbf{D}_k^- (\mathbf{U}_k^-)^T &= \begin{bmatrix} \mathbf{U}_k^- & \mathbf{0}_{n \times m} \end{bmatrix} \begin{bmatrix} (\mathbf{T}_{1k}^-)^T \tilde{\mathbf{D}}_k^- (\mathbf{T}_{1k}^-) & \mathbf{0}_{(n+m) \times (n+m)} \\ \mathbf{0}_{(n+m) \times (n+m)} & (\mathbf{T}_{2k}^-)^T \tilde{\mathbf{D}}_k^- (\mathbf{T}_{2k}^-) \end{bmatrix} \begin{bmatrix} (\mathbf{U}_k^-)^T \\ \mathbf{0}_{m \times n} \end{bmatrix} \\ &= \mathbf{U}_k^- \left[ (\mathbf{T}_{1k}^-)^T \tilde{\mathbf{D}}_k^- (\mathbf{T}_{1k}^-) \right] (\mathbf{U}_k^-)^T, \end{aligned}$$

<sup>a</sup>The “modified” aspect refers to the fact that standard Gram-Schmidt orthogonalization can result in basis vectors that are slightly non-orthogonal due to round-off errors. The resulting vectors are modified [16, Appendix B] to ensure complete orthogonality. The “weighted” aspect refers to the fact that the inner product of two vectors  $\langle \mathbf{u}, \mathbf{v} \rangle$  is defined to be orthogonal with respect to  $\tilde{\mathbf{D}}_k^-$ . Thus  $\langle \mathbf{u}, \mathbf{v} \rangle \triangleq \mathbf{u}^T \tilde{\mathbf{D}}_k^- \mathbf{v}$ .

which is the desired result. To summarize,  $\mathbf{U}_k^-$  and  $\mathbf{D}_k^-$  can be found using

$$\mathbf{Y}_k^-(\mathbf{T}_k^-)^{-T} = \begin{bmatrix} \mathbf{U}_k^- & \mathbf{0}_{n \times m} \end{bmatrix}$$

$$\mathbf{D}_k^- = (\mathbf{T}_{1_k}^-)^T \tilde{\mathbf{D}}_k^- (\mathbf{T}_{1_k}^-).$$

For the update stage of the UDU-MEKF, as with the SR-MEKF, the update for the mean remains as given in Eq. (17), with the caveat that the Kalman gain is computed differently. The covariance update, however, differs in that the factors  $\mathbf{U}_k^-$  and  $\mathbf{D}_k^-$  are updated to yield  $\mathbf{U}_k^+$  and  $\mathbf{D}_k^+$ , instead of working with the covariance directly. Updating  $\mathbf{U}$  and  $\mathbf{D}$  requires measurements to be processed as scalars (i.e. one at a time). The UDU measurement update is done using the Carlson rank-1 update algorithm.<sup>17b</sup> Computationally, the Carlson algorithm performs significantly better than using Joseph's form of the covariance update equation [18, p. 205], and on-par with the standard MEKF covariance update in Eq. (18).

This development summarizes D'Souza [16, Sec. 4 & Appendix C], and the reader is directed there for further details. Assuming a scalar measurement, such that  $\mathbf{H}_k$  is  $1 \times n$  and  $\mathbf{R}_k$  is  $1 \times 1$ , first define

$$\mathbf{f}_k = (\mathbf{U}_k^-)^T \mathbf{H}_k^T$$

$$\mathbf{v}_k = \mathbf{D}_k^- \mathbf{f}_k$$

$$\alpha_k = \mathbf{H}_k \mathbf{U}_k^- \mathbf{v}_k + \mathbf{R}_k,$$

where  $\mathbf{f}_k$  and  $\mathbf{v}_k$  are  $n \times 1$  and  $\alpha_k$  is a scalar. Substituting these definitions and the UDU factorization into the covariance update of Eq. (18), it can be shown that

$$\mathbf{U}_k^+ \mathbf{D}_k^+ (\mathbf{U}_k^+)^T = \mathbf{U}_k^- \underbrace{\left[ \mathbf{D}_k^- - \frac{1}{\alpha_k} \mathbf{v}_k \mathbf{v}_k^T \right]}_{\triangleq \tilde{\mathbf{U}}_k \tilde{\mathbf{D}}_k \tilde{\mathbf{U}}_k^T} (\mathbf{U}_k^-)^T.$$

Therefore, it immediately follows that

$$\mathbf{U}_k^+ = \mathbf{U}_k^- \tilde{\mathbf{U}}_k \quad \text{and} \quad \mathbf{D}_k^+ = \tilde{\mathbf{D}}_k,$$

where, letting  $v_i$  be the  $i^{\text{th}}$  component of  $\mathbf{v}_k$ ,  $\tilde{\mathbf{U}}_k$  is given by

$$\tilde{\mathbf{U}}_k = \begin{bmatrix} 1 & \lambda_2 v_1 & \lambda_3 v_1 & \cdots & \lambda_n v_1 \\ 0 & 1 & \lambda_3 v_2 & \cdots & \lambda_n v_2 \\ 0 & 0 & 1 & \cdots & \lambda_n v_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

In the expression for  $\tilde{\mathbf{U}}_k$ ,

$$\lambda_j \triangleq -\frac{f_j}{\alpha_{j-1}},$$

where  $f_j$  is the  $j^{\text{th}}$  component of  $\mathbf{f}_k$  and, for  $j = 2, \dots, n$ ,  $\alpha_j$  is determined from

$$\alpha_j = \alpha_{j-1} + v_j f_j, \quad \alpha_1 = \mathbf{R}_1 + v_1 f_1.$$

Next, recalling that  $\mathbf{D}_k^+$  is a diagonal matrix, the  $j^{\text{th}}$  diagonal element  $d_j^+$  is given by

$$d_j^+ = \tilde{d}_j = d_j^- \left( \frac{\alpha_{j-1}}{\alpha_j} \right), \quad d_1^+ = d_1^- \left( \frac{\mathbf{R}_1}{\alpha_1} \right)$$

---

<sup>b</sup>Carlson is used rather than Agee-Turner because of how the covariance evolves. A time update *grows* the covariance, and therefore involves a "+" sign ( $\mathbf{P}_k^+ = \mathbf{P}_{k-1}^- + \dots$ ). A measurement update, in contrast, *shrinks* the covariance, and therefore involves a "-" sign ( $\mathbf{P}_k^+ = \mathbf{P}_k^- - \dots$ ). The Agee-Turner rank-1 update uses the form  $\mathbf{P}^+ = \mathbf{P}^- + k \mathbf{v} \mathbf{v}^T$ , in which  $k$  is the variance of the residuals and  $\mathbf{v}$  is the gain, and  $\mathbf{v}$  is not required to be the optimal (i.e. Kalman) gain  $\mathbf{K}$ . For a measurement update, this form will be subtractive rather than additive in nature, and can lead to numerical issues when updating  $\mathbf{P}$ . The Carlson update avoids these issues by assuming that  $\mathbf{v}$  is in fact the optimal gain  $\mathbf{K}$ , and is therefore used for the UDU measurement update.

Finally, recall that  $\mathbf{K}_k$  is required for the mean update given by Eq. (17); for the UDU-MEKF,  $\mathbf{K}_k$  is given by

$$\mathbf{K}_k = \tilde{\mathbf{K}}_n / \alpha_n, \quad \tilde{\mathbf{K}}_j = \tilde{\mathbf{K}}_{j-1} + v_j (\mathbf{U}_k^-)^{(j)}, \quad \tilde{\mathbf{K}}_1 = \begin{bmatrix} v_1 & 0 & \cdots & 0 \end{bmatrix}^T,$$

where  $(\mathbf{U}_k^-)^{(j)}$  is the  $j^{\text{th}}$  column of  $\mathbf{U}_k^-$ .

The update described here assumes scalar measurement processing; that is, given a  $p$ -dimensional measurement, each scalar element is considered one at a time, and the preceding algorithm is executed for each available scalar measurement. Provided that the measurement noise covariance matrix is diagonal, this can be performed without modification. When the measurement noise covariance matrix is not diagonal, a de-correlation step must first be performed to transform the received measurements into new measurements whose measurement noise covariance matrix is diagonal. Additionally, since  $p$  scalar updates are performed, there are  $p$  computations for  $\Delta \mathbf{x}_k$  that are determined via Eq. (19). These are all added together to aggregate the effects of the  $p$  measurements before employing the mean update of Eq. (17).

## V. Results and Discussion

In this section, simulations of a lunar lander are used to assess any differences in performance between the full covariance, UDU, and Cholesky square-root formulations of the MEKF. The following discussion analyzes the single run performance of each filter formulation for each estimated state variable (position, velocity, and attitude) and parameter (accelerometer bias, gyroscope bias, altimeter bias, and star camera bias). It also presents a 1000-trial Monte Carlo analysis of each filter formulation (where all noises are resampled in each trial), which is used to evaluate the statistical consistency of a single filter run.

Accordingly, consider the representative lunar descent trajectory depicted in Fig. 1. The simulated lunar lander is equipped with an IMU, a spherical altimeter, and a quaternion star camera that follow the previously described models, with their statistics described in Table 1<sup>c</sup>. Note that the altimeter noise is

**Table 1. Sensor configuration for the simulated lander.**

Sensor	Rate	Noise ( $1\sigma$ )	Bias ( $1\sigma$ )	When active?
Accelerometer	40 Hz	$35\mu g/\sqrt{\text{Hz}}$	$300\mu g$	Always
Gyroscope	40 Hz	$< 0.07^\circ/\sqrt{\text{hr}}$	$1^\circ/\text{hr}$	Always
Altimeter	10 Hz	$[500, 5]$ m	0.5 m	alt. $< 15,000$ m
Star Camera	1 Hz	30 arc-seconds	10 arc-seconds	When <i>not</i> thrusting

described as  $[500, 5]$  meters. This means that the measurement noise starts at 500 meters ( $1\sigma$ ) when the altimeter is activated at an altitude of 15,000 meters, and decreases linearly with altitude to 5 meters as the altitude decreases. This is done to coarsely model the noise behavior commonly exhibited by altimeters. Note further that the star camera is assumed to provide measurements only when the vehicle is quiescent, due to the vibrations induced when the vehicle is thrusting. All noises are assumed to be zero-mean Gaussian for simplicity, but this assumption is not required for any of methods described in this paper. For all filters, underweighting is applied during each measurement processing step.

Given the true initial position, velocity, and attitude from the trajectory, denoted as  $\mathbf{r}_0$ ,  $\mathbf{v}_0$  and  $\bar{\mathbf{q}}_0$ , all filters are initialized in the same fashion. First, the initial covariance for  $\mathbf{r}_0$ ,  $\mathbf{v}_0$  and attitude error,  $\Delta \phi_0$  (corresponding to  $\bar{\mathbf{q}}_0$ ), are defined to be

$$\mathbf{P}_{r,v,q}(t_0) = \text{diag}(1000^2, 1000^2, 1000^2, 0.1^2, 0.1^2, 0.1^2, 0.573^2, 0.573^2, 0.573^2)$$

in (meters)<sup>2</sup>, (meters per second)<sup>2</sup>, and (degrees)<sup>2</sup>, respectively, where  $\mathbf{P}_{r,v,q}(t_0)$  is meant to denote the initial uncertainties in position, velocity, and attitude error at  $t_0$ . Then, initial values of  $\hat{\mathbf{r}}_0$ ,  $\hat{\mathbf{v}}_0$  and  $\Delta \hat{\phi}_0$

<sup>c</sup>These values are from the specifications sheet for the Northrop Grumman LN-200S IMU, which provides specifications in terms of accelerations and body rates. A units conversion is required to produce values appropriate for integrated accelerations,  $\Delta \mathbf{v}$ , and body rates,  $\Delta \theta$ . For these values, see: <http://www.northropgrumman.com/Capabilities/LN200F0G/Documents/ln200s.pdf>

for each filter trial are drawn according to a Gaussian distribution with a mean given by the true position, velocity, and attitude and covariance  $\mathbf{P}_{r,v,q}(t_0)$ . An initial quaternion estimate is recovered from this attitude error via

$$\hat{\mathbf{q}}_0 = \begin{bmatrix} \frac{1}{2}\Delta\hat{\phi}_0 \\ 1 \end{bmatrix} \otimes \bar{\mathbf{q}}_0,$$

which is subsequently normalized to ensure a valid attitude quaternion. Then, using the statistics in Table 1, true values of the accelerometer, gyroscope, altimeter, and star camera biases are drawn according to Gaussian distributions parameterized by the listed statistics. The initial estimates of these statistics are taken to be the values in the table. Initialization is completed by concatenating  $\hat{\mathbf{r}}_0$ ,  $\hat{\mathbf{v}}_0$ ,  $\hat{\mathbf{q}}_0$ , and the initial bias mean estimates to form  $\mathbf{m}_0$ , and  $\mathbf{P}_{r,v,q}(t_0)$  is concatenated (block diagonally) with the initial covariances of each of the biases to form  $\mathbf{P}_0$ .

The aforementioned single-run comparisons of the MEKF, UDU-MEKF, and SR-MEKF are shown in Figs. 2–8, where position and velocity are presented in the UVW frame. These figures show the error and covariance histories (presented as  $3\sigma$  intervals) for all state and parameter estimates as the vehicle descends. The large “snap down” in errors toward the end of the simulation corresponds to when the altimeter is activated and permitted to collect data. Note that before this time, the vehicle relies solely upon the discrete dead-reckoning equations to provide estimates of the states and parameters that the star camera cannot measure (e.g. position, velocity, etc.). This explains the continual growth in the state uncertainty until the end of the simulation, at which point the filter is provided with altimeter data.

All filters produce errors that are within their respective  $3\sigma$  intervals, with the exception of the attitude errors during the terminal descent. In this case, all filters are identically plagued with a combination sampled IMU data during powered flight (i.e. thrusting) and linearization errors. The effect of sampled IMU data during powered flight, in which the vehicle experiences large accelerations and body rates, takes the form of integration errors in the discrete dead-reckoning equations. These errors accumulate in both the position and attitude states, and since these state variables are closely correlated, the filter is unable to appropriately quantify their error statistics. Linearization errors also grow during this period, as the conditional mean estimate of the state drifts further from the true state value. Due to the thrusting during terminal descent, the star camera is unable to collect attitude information with which to help the filter correct these errors, and these errors accumulate as biases in the attitude estimates. This undesirable performance could be mitigated by the use of higher rate IMU data; however, note that the attitude errors, while not well statistically quantified, remain quite low at about half a degree, at their worst.

Figure 7 indicates that the altimeter bias is unobservable given the total lack of change in the errors or covariance values. This is due to the altimeter being collocated with IMU, and displacing this sensor would allow for attitude correlations to build. In this case, one would expect useful updates to the altimeter bias. Also, while quaternion attitude measurements are provided by the star camera during most of the descent, Fig. 8 demonstrates that a vehicle pitching maneuver at about 1,400 seconds MET allows correlations to accumulate, thereby permitting an update to the star camera bias.

Figures 9–11 represent a subset of the Monte Carlo analysis. Here, position, attitude, and gyro bias are represented in lieu of all the state variables and parameters for compactness, and these plots demonstrate most effectively any differences between the MEKF, UDU-MEKF, and SR-MEKF methods. This portion of the analysis seeks to uncover differences between the error covariance provided by the Monte Carlo analysis and the error covariance provided by a single filter run. Doing so elucidates how well the filters quantify uncertainties during descent. The position results in Fig. 9 indicate general agreement between the three methods, but they are not without their differences. Two primary differences are immediately apparent in this case. First, the SR-MEKF demonstrates a conservative behavior in the position states with respect to Monte Carlo prior to the altimeter collecting data. This indicates that the prediction stage of the SR-MEKF provides conservative covariance estimates. Second, the UDU-MEKF is actually overly optimistic, and the covariance provided by the single run is below that of the Monte Carlo results. This is perhaps more alarming than the conservative covariance provided by the SR-MEKF due to the risks of over-convergence in EKF implementations. The attitude results in Fig. 10 indicate that the UDU-MEKF and SR-MEKF agree with the Monte Carlo trends, but that the full covariance MEKF actually provides overly confident covariance estimates as well. Just as with the UDU-MEKF’s over-confidence in the position states, the MEKF fails to appropriately represent the statistics of its errors. Figure 11 demonstrates that all three filter mechanizations produce equally consistent error covariances for parameters with respect to the Monte Carlo analysis.

The results in Figs. 12 and 13 are enhanced, grid-view counterparts of the position and attitude single run/Monte Carlo results (Figs. 9 and 10) during terminal descent. The pertinent conclusions are as follows. In the position states, the MEKF and SR-MEKF tend toward being conservative, whereas the UDU-MEKF matches more closely at certain points, while becoming overly confident at other points. In attitude, the performance of the MEKF indicates overly confident estimates, where UDU-MEKF and SR-MEKF perform quite similarly. Drawing conclusions with regard to statistics in attitude is perhaps challenging (see previous discussion on Fig. 4), but comparison to the Monte Carlo analysis can at least add valuable insight.

It should be noted that, due to loss of positive definiteness, which results in algorithm failure, a small number of the 1000 Monte Carlo trials for the MEKF have to be excluded from the data pool. This does not occur with either the UDU-MEKF or the SR-MEKF implementations.

## VI. Conclusion

A comparison of different MEKF formulations for navigation is presented, including methods that are formulated to use the full covariance, UDU factors, and Cholesky square-root factors. These are presented within the context of navigation for planetary landers, and the salient features of the three algorithms are investigated in a simulation. Using a simulated lunar descent trajectory, inertial navigation is performed with a strapdown IMU, and external measurements in the form of altimeter and star camera data are processed to improve the navigation solution. A Monte Carlo analysis is presented to assess the statistical consistency of all three methods.

The results indicate that while all three filter formulations perform similarly, some undesirable characteristics do appear. First, the full covariance formulation of the MEKF exhibits failure in a subset of the Monte Carlo trials due to loss of positive definiteness of its covariance matrix. Second, each filter formulation exhibits some level of inconsistency in its covariance estimate. The MEKF produces inconsistent (over confident) covariance estimates in the attitude states (such as demonstrated in Fig. 13). The UDU-MEKF produces over confident covariance estimates in the position states (such as demonstrated in Fig. 9). The SR-MEKF produces inconsistent (conservative) covariance estimates in the position states (as can be seen in Fig. 9). Finally, all filters are able to maintain statistically consistent estimates of the parameters.

Broadly speaking, it is clear that factorized representations of covariance, either by UDU or Cholesky square-root, alleviate the issues faced by full covariance MEKF implementations. In drawing contrasts, the appearance of conservative covariance estimates from the SR-MEKF is more desirable than the underestimated covariance sometimes produced by the UDU-MEKF, though even these differences are small. Selection between the two may also come down to preference based on the specific problem, such as vector vs. scalar processing (SR-MEKF vs. UDU-MEKF) of data, and the computational resources available to the filter.

## VII. Acknowledgments

This work was partially supported by a NASA Space Technology Research Fellowship and through Grant NNX16AF11A. The authors would also like to acknowledge the many helpful discussions with Drs. Chris D'Souza and Renato Zanetti of NASA Johnson Space Center.

## References

- <sup>1</sup>Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME-Journal of Basic Engineering*, Vol. 82, No. Series D, 1960, pp. 35–45.
- <sup>2</sup>Smith, G. L., Schmidt, S. F., and McGee, L. A., "Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity on Board a Circumlunar Vehicle," Technical Report R-135, NASA, 1962.
- <sup>3</sup>Gelb, A., *Applied Optimal Estimation*, M.I.T. Press, 1974.
- <sup>4</sup>Zanetti, R., DeMars, K. J., and Bishop, R. H., "Underweighting Nonlinear Measurements," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 5, 2010, pp. 1670–1675.
- <sup>5</sup>Horn, R. A. and Johnson, C. R., *Matrix Analysis*, Cambridge University Press, Cambridge, United Kingdom, 1985.
- <sup>6</sup>Bierman, G. J., *Factorization Methods for Discrete Sequential Estimation*, Vol. 1, Academic Press, 1977.
- <sup>7</sup>Maybeck, P. S., *Stochastic Models, Estimation, and Control*, Vol. 1, Academic Press, 1979.
- <sup>8</sup>Zanetti, R., *Advanced Navigation Algorithms for Precision Landing*, Ph.D. thesis, The University of Texas at Austin, 2007.
- <sup>9</sup>Savage, P. G., *Strapdown Analytics*, Strapdown, Maple Plain, MN.

- <sup>10</sup>Woodman, O. J., *An Introduction to Inertial Navigation*, Tech. Rep. UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, August 2007.
- <sup>11</sup>Markley, F. L. and Crassidis, J. L., *Fundamentals of Spacecraft Attitude Determination and Control*, Springer, 2014.
- <sup>12</sup>Crassidis, J. L. and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, CRC press, 2011.
- <sup>13</sup>McCabe, J. S. and DeMars, K. J., “Particle Filter Methods for Space Object Tracking,” Proceedings of the AIAA Astrodynamics Specialists Conference, San Diego, CA, 2014.
- <sup>14</sup>van der Merwe, R., *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*, Ph.D. thesis, Oregon Health and Science University, Portland, Oregon, 2004.
- <sup>15</sup>Golub, G. H. and Loan, C. F. V., *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 3rd ed., 1996.
- <sup>16</sup>D’Souza, C., “A Primer on the Orion Absolute Navigation UDU Filter,” Technical Brief EG-DIV-11-24, NASA Johnson Space Center, Engineering Directorate, February 28, 2011.
- <sup>17</sup>Carlson, N. A., “Fast Triangular Formation of the Square Root Filter,” *AIAA Journal*, Vol. 11, No. 9, September, 1973, pp. 1259–1265.
- <sup>18</sup>Tapley, B. D., Schutz, B. E., and Born, G. H., *Statistical Orbit Determination*, Elsevier Academic Press, New York, NY, 2004.

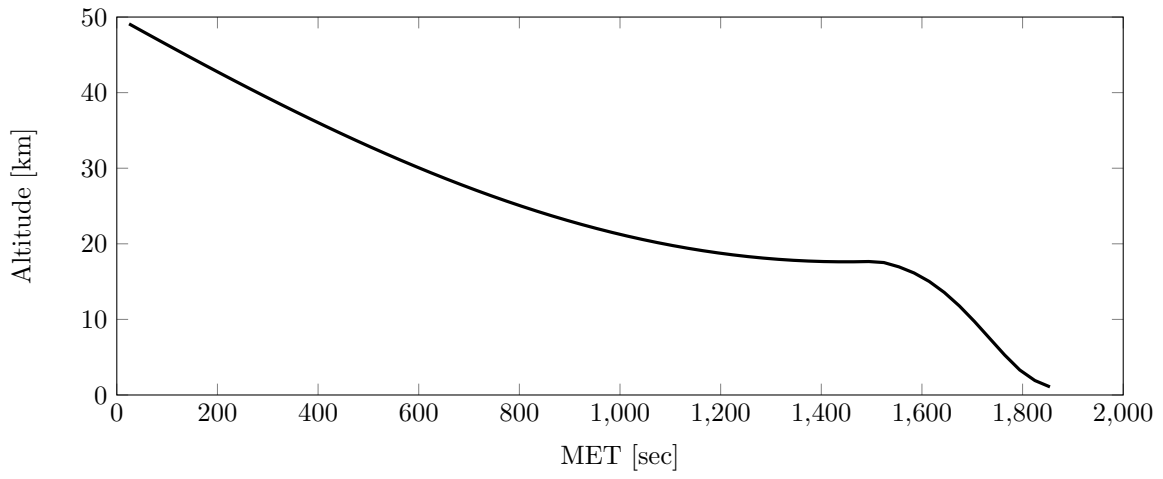


Figure 1. Depiction of the simulation descent trajectory altitude vs. mission elapsed time (MET).

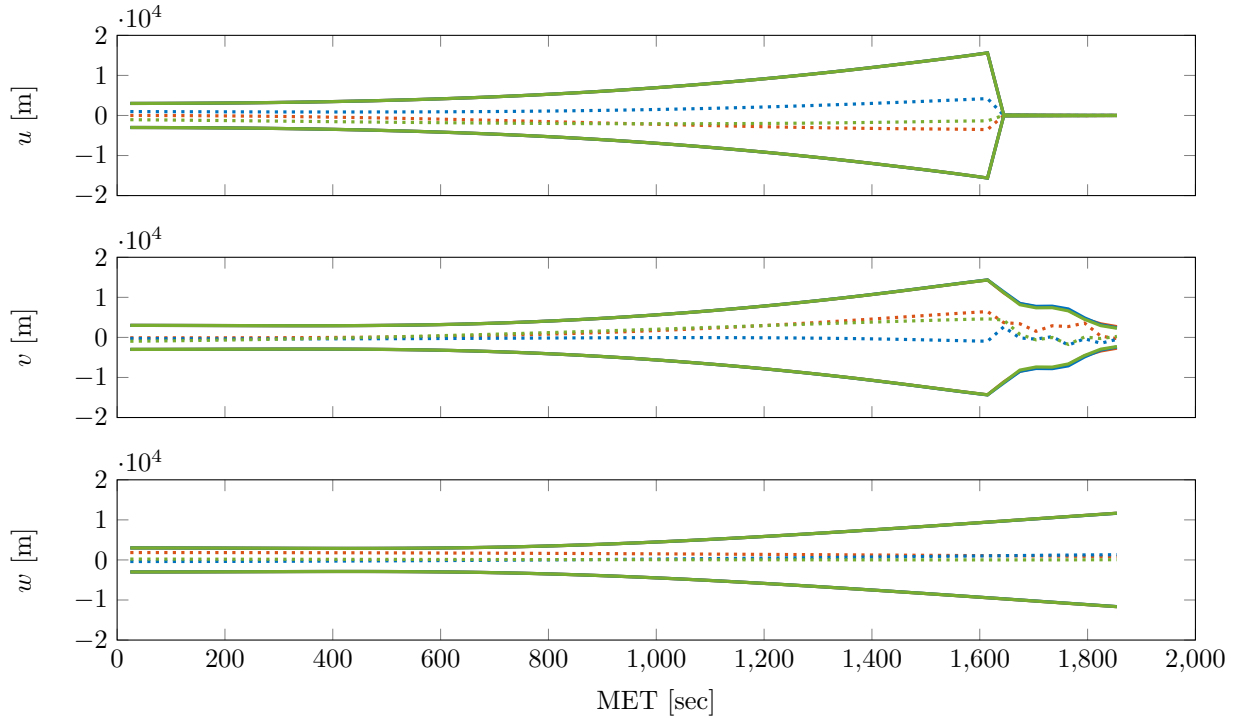


Figure 2. Position errors and  $3\sigma$  intervals (dotted and solid lines, respectively) for the full covariance, UDU, and Cholesky square-root formulations (orange, blue, and green, respectively).

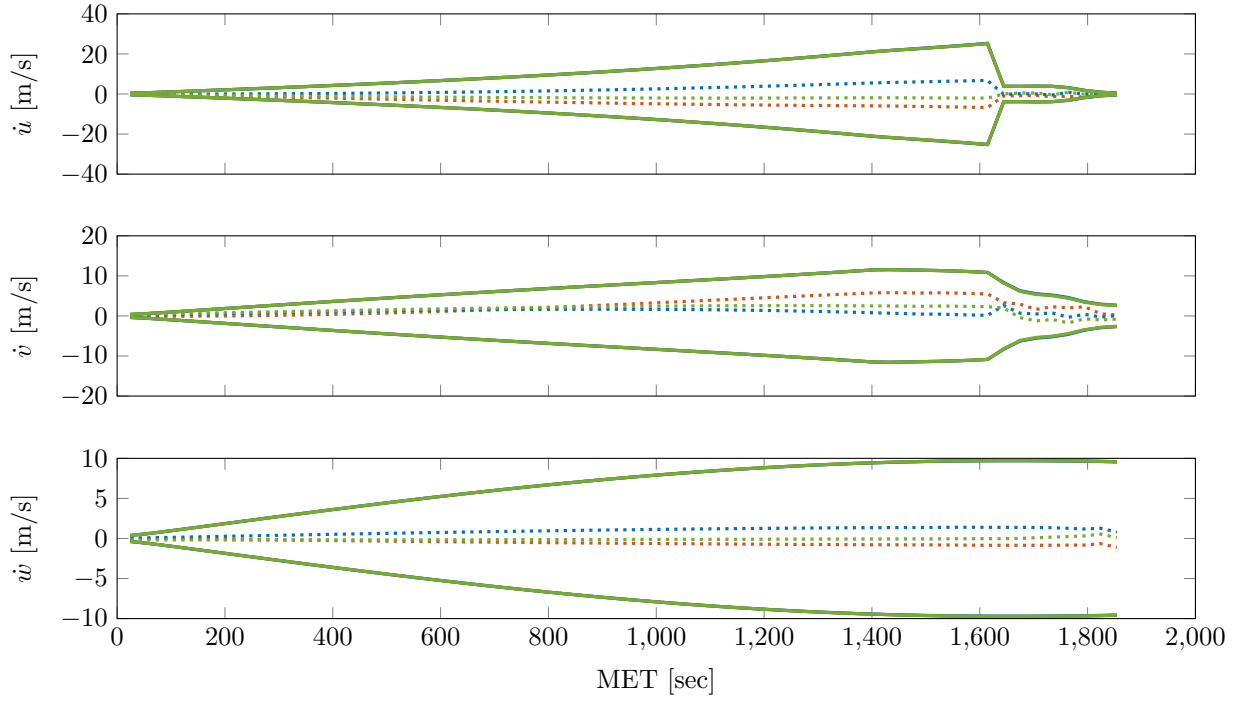


Figure 3. Velocity errors and  $3\sigma$  intervals (dotted and solid lines, respectively) for the full covariance, UDU, and Cholesky square-root formulations (orange, blue, and green, respectively).

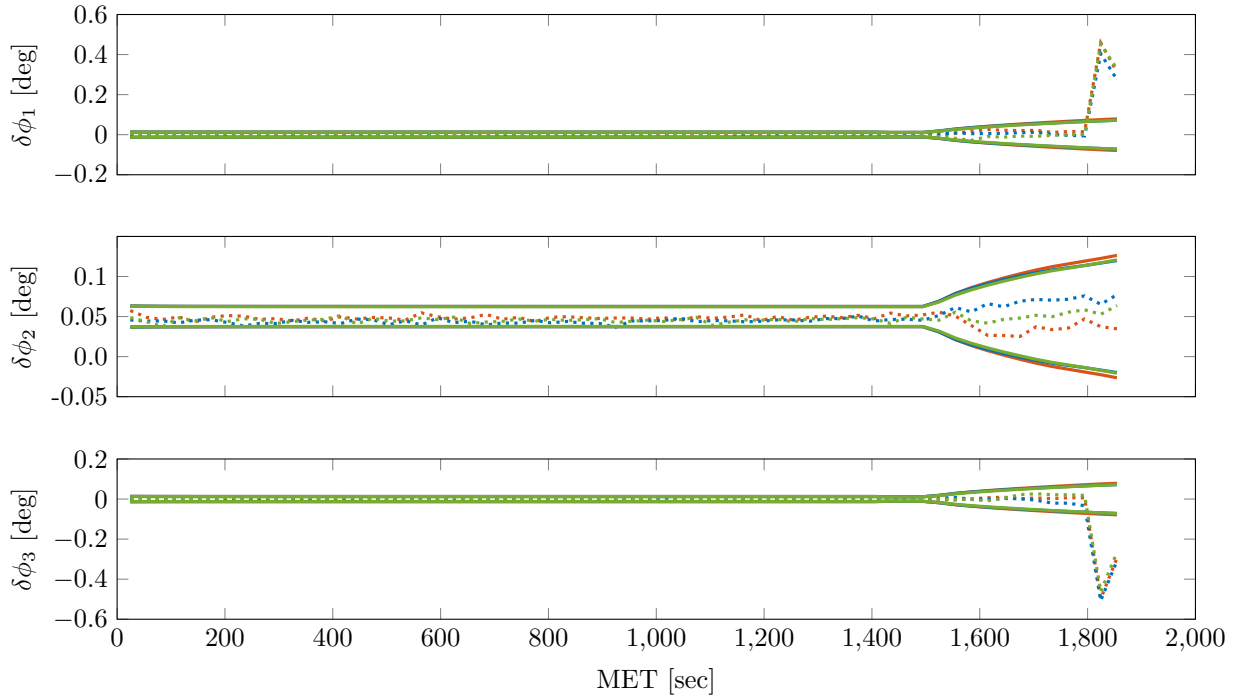


Figure 4. Attitude errors and  $3\sigma$  intervals (dotted and solid lines, respectively) for the full covariance, UDU, and Cholesky square-root formulations (orange, blue, and green, respectively).



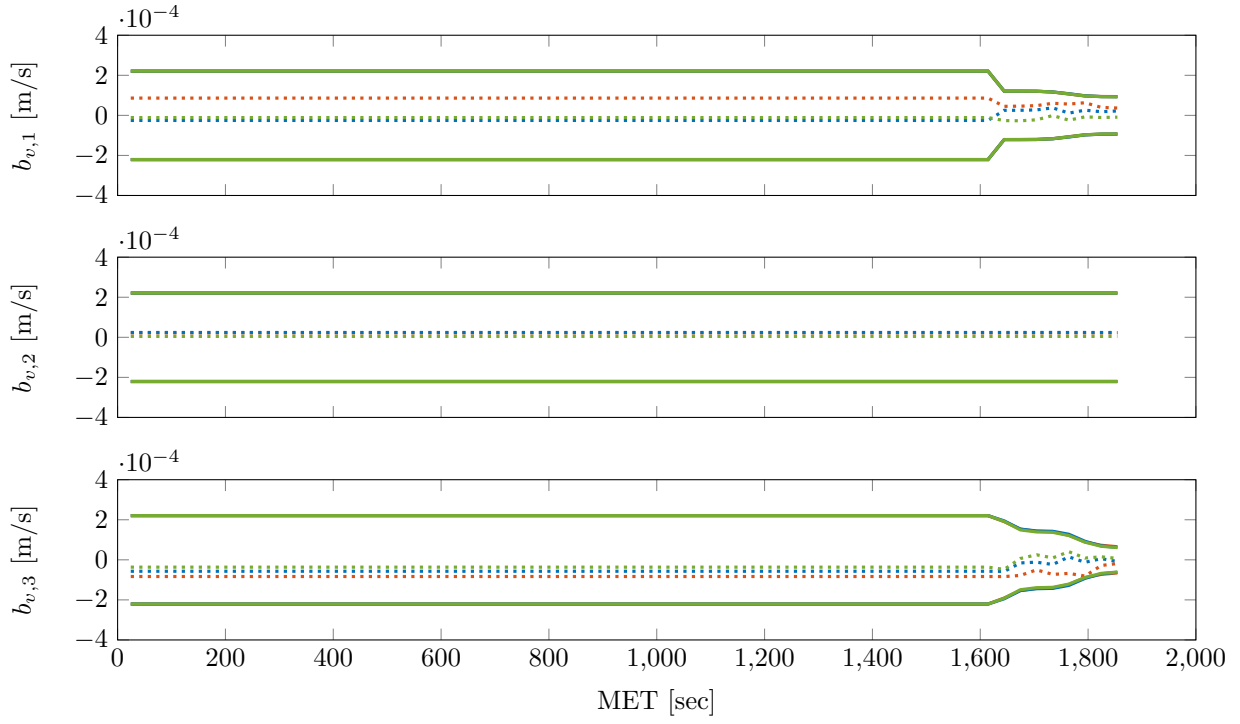


Figure 5. Accelerometer bias errors and  $3\sigma$  intervals (dotted and solid lines, respectively) for the full covariance, UDU, and Cholesky square-root formulations (orange, blue, and green, respectively).

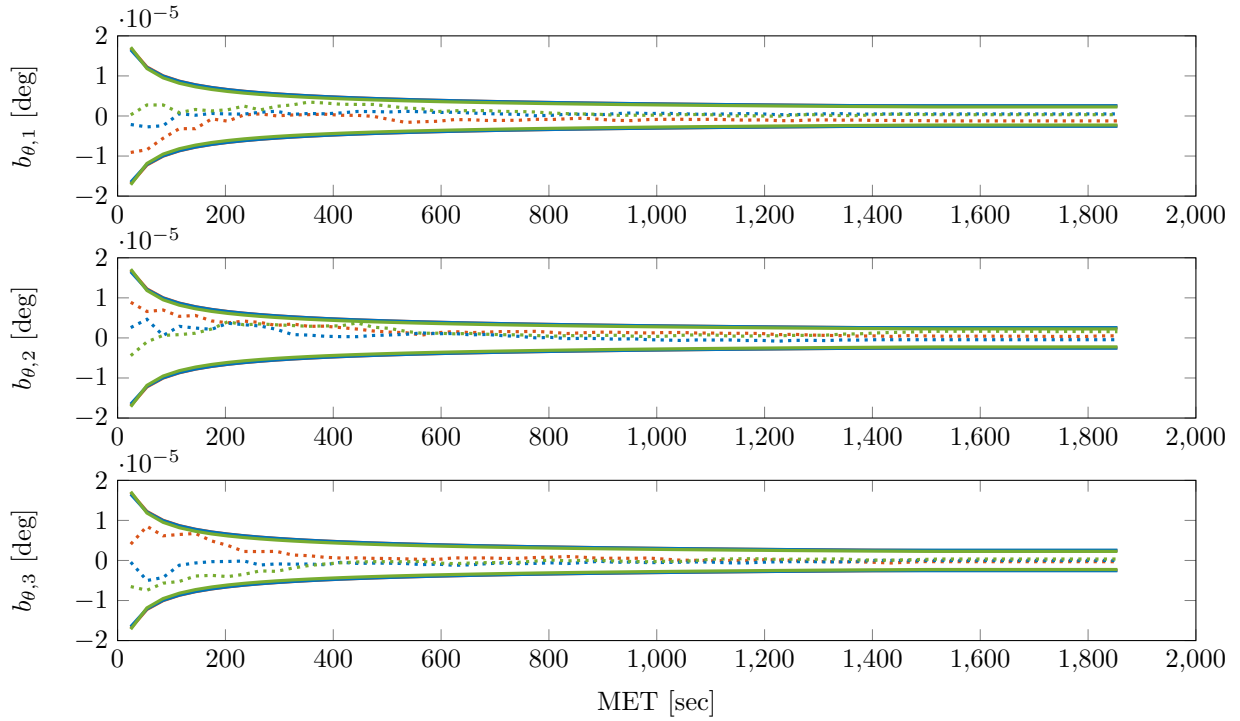


Figure 6. Gyroscope bias errors and  $3\sigma$  intervals (dotted and solid lines, respectively) for the full covariance, UDU, and Cholesky square-root formulations (orange, blue, and green, respectively).

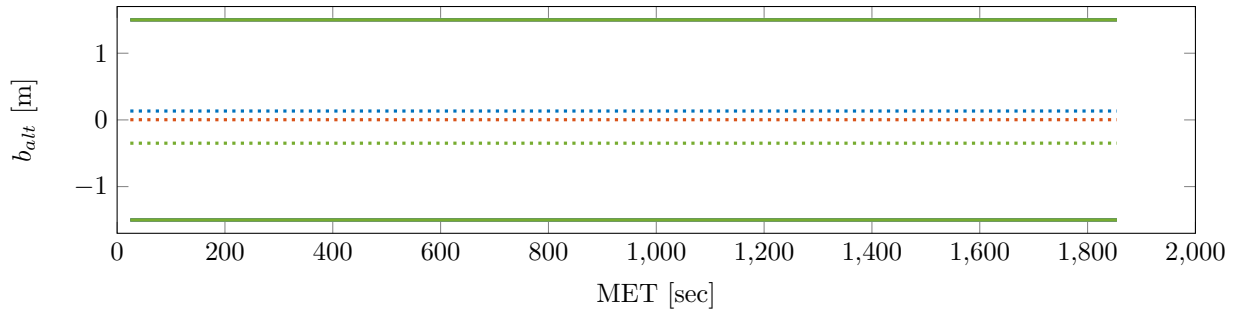


Figure 7. Altimeter bias errors and  $3\sigma$  intervals (dotted and solid lines, respectively) for the full covariance, UDU, and Cholesky square-root formulations (orange, blue, and green, respectively).

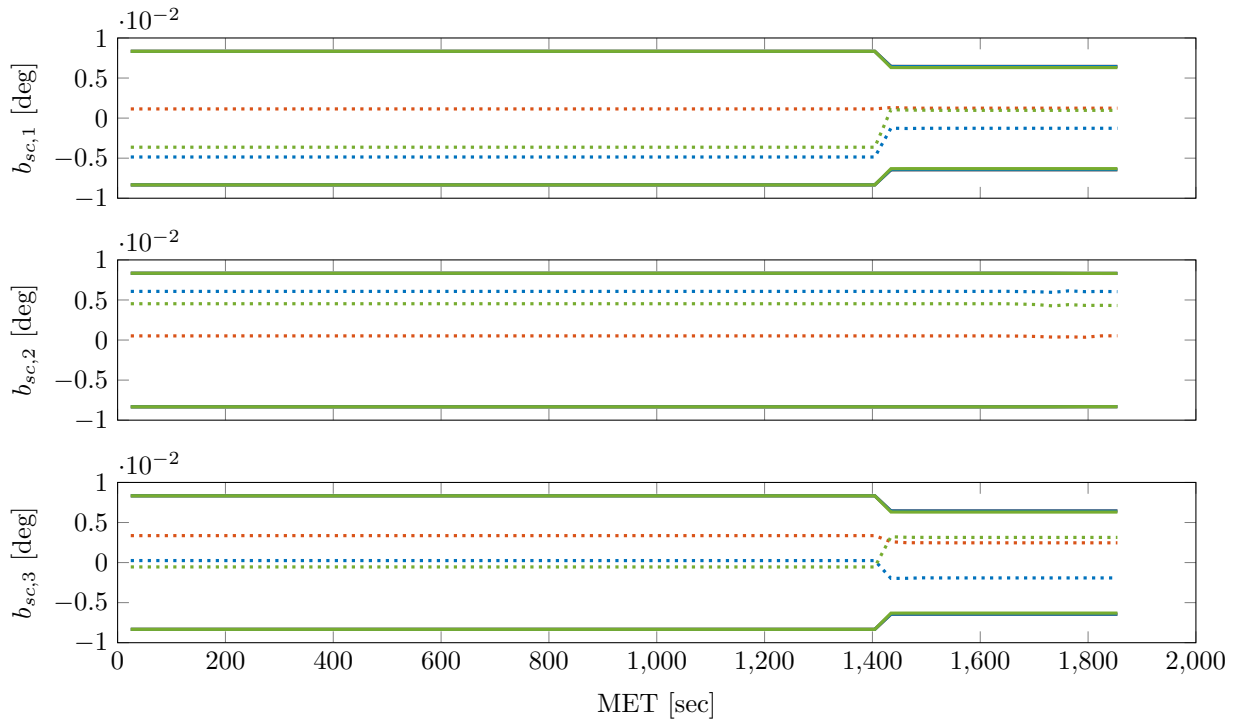


Figure 8. Star camera bias errors and  $3\sigma$  intervals (dotted and solid lines, respectively) for the full covariance, UDU, and Cholesky square-root formulations (orange, blue, and green, respectively).

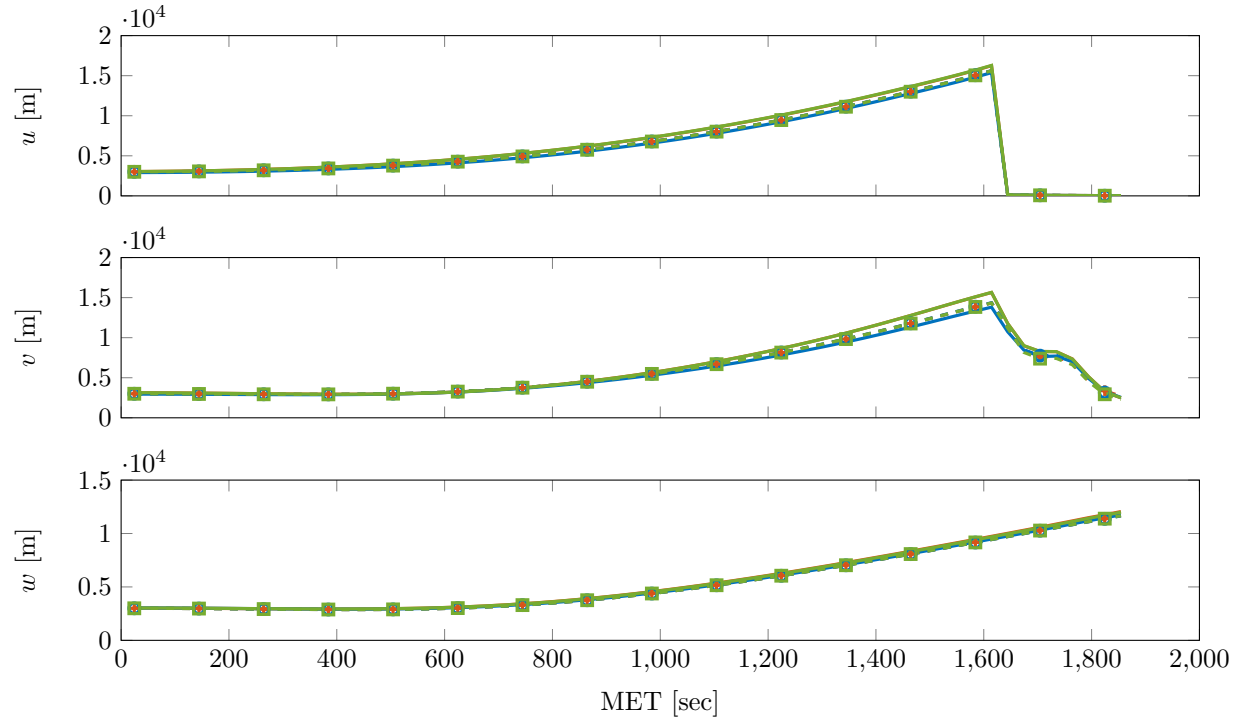


Figure 9. Position  $3\sigma$  intervals for a single run and Monte Carlo trials (solid and dashed, marked lines, respectively) for the full covariance, UDU, and Cholesky square-root formulations (orange/ $\times$ , blue/circle, and green/square, respectively).

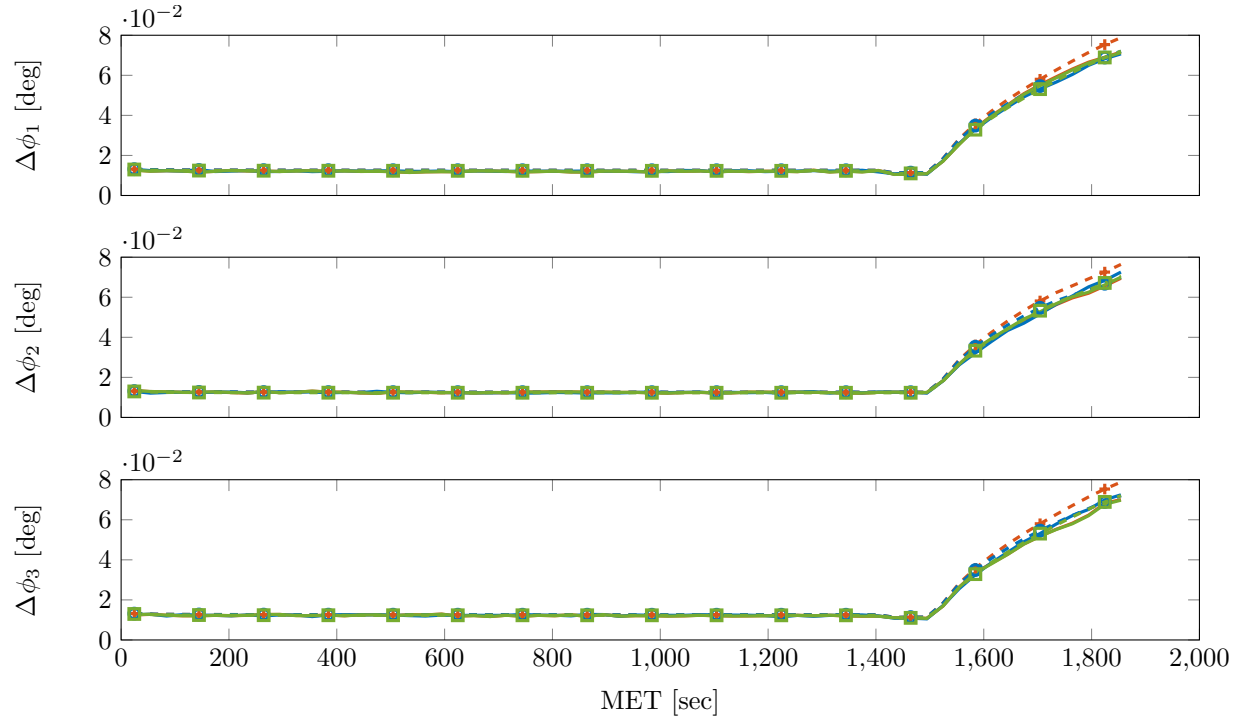


Figure 10. Attitude  $3\sigma$  intervals for a single run and Monte Carlo trials (solid and dashed, marked lines, respectively) for the full covariance, UDU, and Cholesky square-root formulations (orange/ $\times$ , blue/circle, and green/square, respectively).

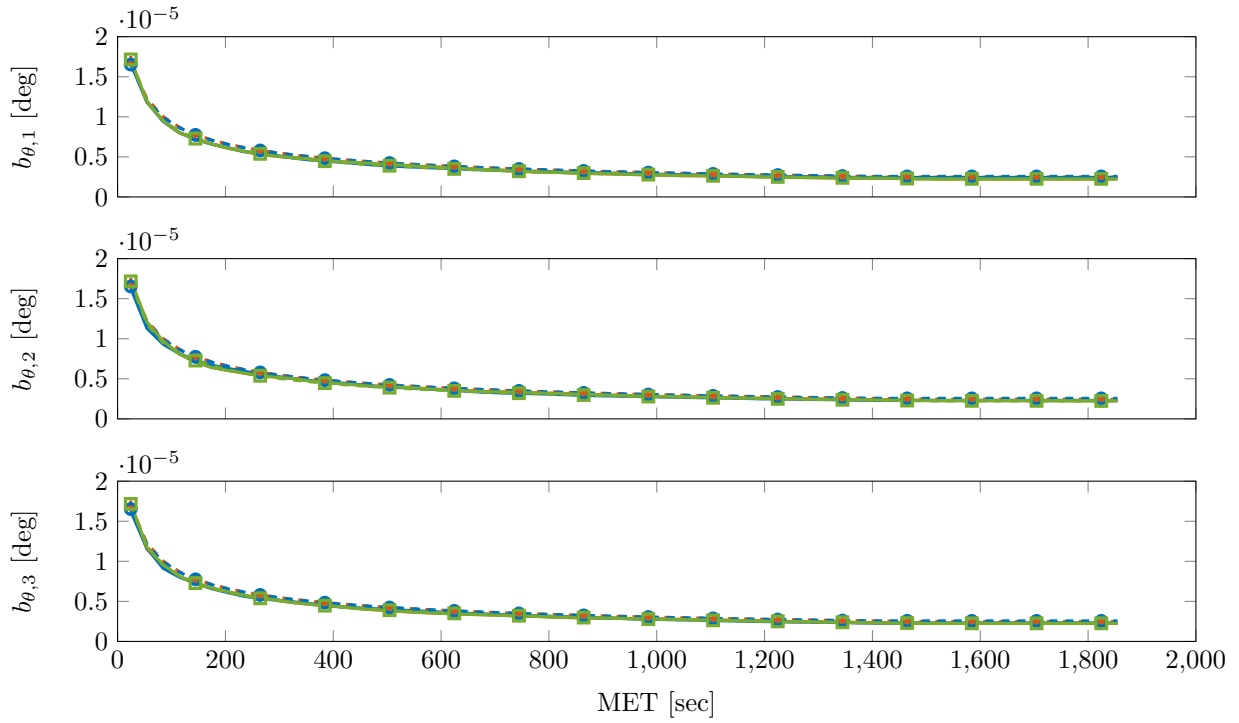


Figure 11. Gyroscope bias  $3\sigma$  intervals for a single run and Monte Carlo trials (solid and dashed, marked lines, respectively) for the full covariance, UDU, and Cholesky square-root formulations (orange/ $\times$ , blue/circle, and green/square, respectively).

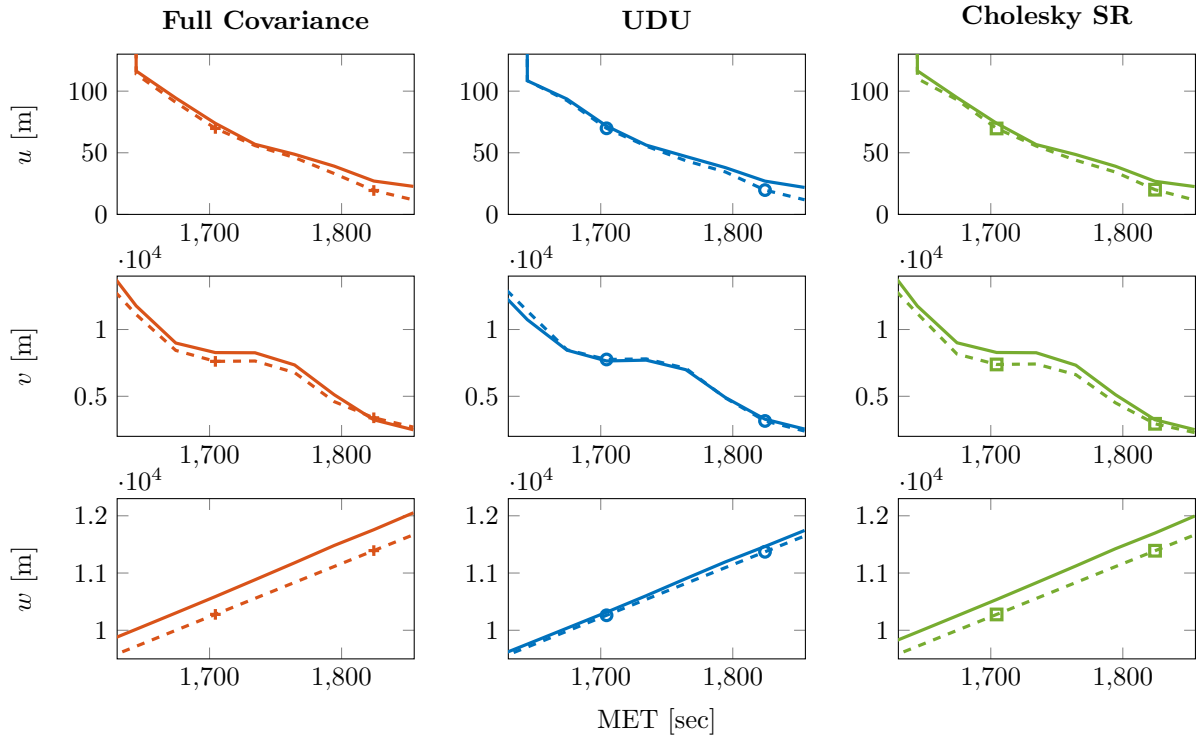


Figure 12. Terminal descent position  $3\sigma$  intervals for a single run and Monte Carlo trials (solid and dashed, marked lines, respectively) for the full covariance, UDU, and Cholesky square-root formulations (orange/ $\times$ , blue/circle, and green/square, respectively).

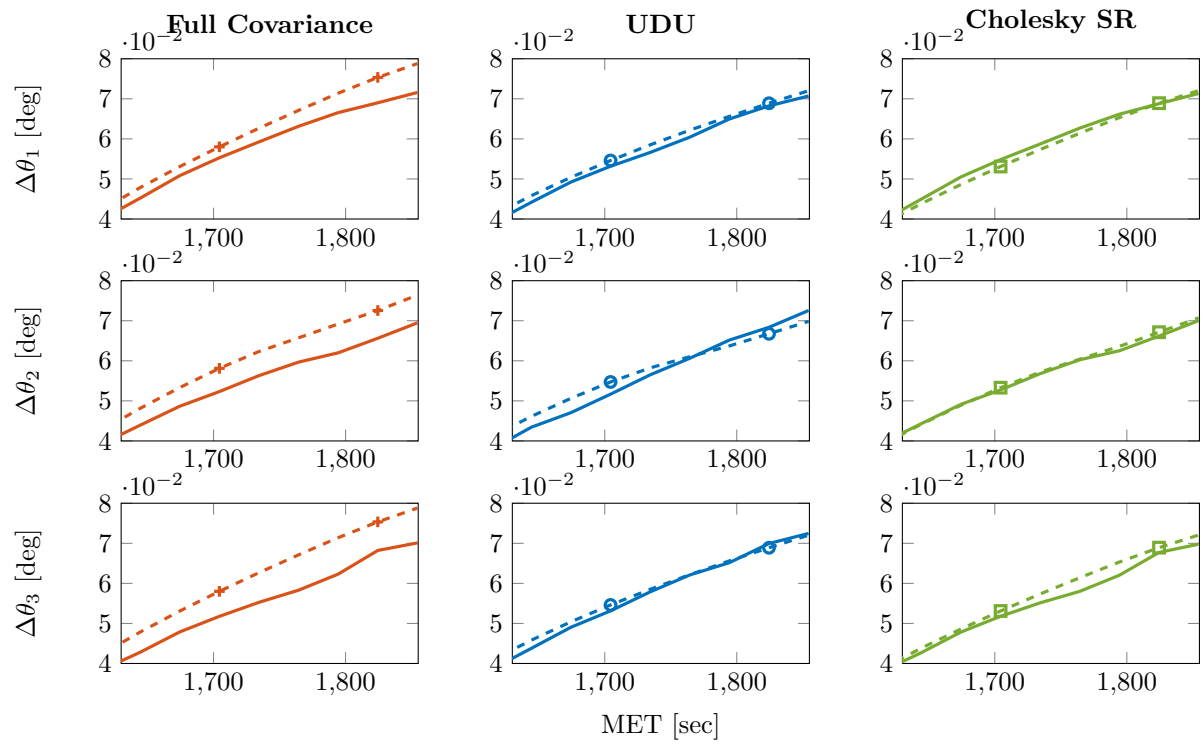


Figure 13. Terminal descent attitude  $3\sigma$  intervals for a single run and Monte Carlo trials (solid and dashed, marked lines, respectively) for the full covariance, UDU, and Cholesky square-root formulations (orange/ $\times$ , blue/circle, and green/square, respectively).